Domain Adaptation in Semantic Role Labeling using a Neural Language Model and Linguistic Resources

Quynh Thi Ngoc Do, Steven Bethard and Marie-Francine Moens

***Abstract*—We propose a method for adapting Semantic Role Labeling (SRL) systems from a source domain to a target domain by combining a neural language model and linguistic resources to generate additional training examples. We primarily aim to improve the results of Location, Time, Manner and Direction roles. In our methodology, main words of selected predicates and arguments in the source-domain training data are replaced with words from the target domain. The replacement words are generated by a language model and then filtered by several linguistic filters (including Part-Of-Speech (POS), WordNet and Predicate constraints). In experiments on the out-of-domain CoNLL 2009 data, with the Recurrent Neural Network Language Model (RNNLM) and a well-known semantic parser from Lund University, we show enhanced recall and F1 without penalizing precision on the four targeted roles. These results improve the results of the same SRL system without using the language model and the linguistic resources, and are better than the results of the same SRL system that is trained with examples that are enriched with word embeddings. We also demonstrate the importance of using a language model and the vocabulary of the target domain when generating new training examples.**

***Keywords*—Semantic role labeling, open domain, language model, linguistic resources.**

I. Introduction

AN essential requirement for machine understanding of text is the ability to detect the events that are being described and the event participants in text. Semantic Role Labeling is the natural language processing task that aims to solve this problem by recognizing "Who did What to Whom, and How, When and Where?" in text [1]. For example, the processing of the sentence "Mary gave Peter a book at school yesterday" should result in the identification of a "giving" event with "Mary" as the *Agent*, "Peter" as the *Recipient*, "a book" as the *Thing being given*, "at school" as the *Location*, and "yesterday" as the *Time*: "[Mary *Agent*] gave (*giving*) [Peter *Recipient*] [a book *Thing being given*] [at school *Location*] [yesterday *Time*]".

In this paper, we call an event ("giving") in a sentence the *semantic frame*, the verb or noun that evokes the frame ("gave") the *predicate*, the words that play a role in the event ("Mary", "Peter", "a book", "at school", "yesterday") the *arguments* and their roles ("Agent", "Recipient", "Thing being given", "Location", "Time") the *semantic roles*. The task of SRL is to detect the event, to identify its arguments and to assign the correct semantic roles to the arguments. Given the predicates, the available systems can reach an F1[1] score of 85%[2] when the domains of the training and testing data are the same. We witness a significant drop in F1 values when a semantic role labeler is applied on a domain other than the one it is trained on[3]. A large part of semantic meaning resides in the individual words, yet many words in the target domain have never been seen in the source domain training data.

To solve this problem, semi-supervised and unsupervised approaches have been considered as a promising solution since manual annotation is expensive and time consuming. The most generic use of unlabeled examples regards the building of language models, i.e., probabilistic models of language, often in the form of n-gram word models. Recently, we see some attempts to use such language models in a semi-supervised setting for semantic recognition [2], [3]. In most of these settings, other words or a statistical class of words provided by the language model enriches the feature vectors used in training, or they are used to create training examples artificially. The language models offer a kind of weak or distant supervision when training the semantic classifier. In this work, we focus on neural language models (also known as context-predicting semantic vector models) which are the new kids on the distributional semantics block [4]. It has been shown that distributional semantic models which use vectors that keep track of the contextual information provides a good approximation to word meaning, since semantically similar words tend to have similar contextual distributions [4].

However, there is no principled way to use such language information in semantic recognition. In the context of semantic recognition other words are only valid replacements in a sentence or phrase context when they would convey the same semantic role or function as the one we are looking for; otherwise the recognizer will be trained with noisy data. For instance, in the sentence "I went home by midnight", when recognizing temporal expressions, replacing the word "midnight" by the word "car", would not help the training of a semantic recognizer of temporal expressions. However, when training a semantic role classifier of actor in the sentence "The cat likes to drink milk." or "The girl likes to drink milk.", "girl" and "cat" are perfectly exchangeable in the training examples. A second difficulty when using language model information is that the most probable replacement words

Q.T.N. Do is affiliated with the Department of Computer Science, Katholieke Universiteit Leuven, Belgium. Email: quynhngocthi.do@cs.kuleuven.be

S. Bethard is affiliated with the Department of Computer and Information Sciences, University of Alabama at Birmingham, United States. E-mail: bethard@cis.uab.edu

M.F. Moens is affiliated with the Department of Computer Science, Katholieke Universiteit Leuven, Belgium. Email: sien.moens@cs.kuleuven.be

[1]Harmonic mean of recall and precision.

[2]https://ufal.mff.cuni.cz/conll2009-st/results/results.php

[3]https://ufal.mff.cuni.cz/conll2009-st/results/results.php

might already be seen in the training data and would not help to improve the learned model, while valid linguistic expressions can have a low probability but might be useful training candidates. Notwithstanding, unlabeled examples offer a wealth of information that could be leveraged by many semantic recognition tasks.

The goal of this paper is to investigate how to best integrate generic language model information when training an accurate semantic role labeler in order to improve specific semantic roles and to compare and evaluate the models when the trained model is applied to target-domain texts that use different words from the source-domain texts that the model is trained on.

The recognition of circumstance semantic roles like Location, Time etc. is very important to understand the full meaning of an event, while the performance of the current SRL systems on those roles is often very poor, especially in an out-of-domain testing scheme. In this paper, we aim to improve SRL on four PropBank circumstance roles: AM-LOC (Location), AM-TMP (Time), AM-MNR (Manner) and AM-DIR (Direction). We develop a methodology to generate additional training data for SRL by replacing selected words in training examples. For each selected word from the source domain, a list of replacement words which we believe can occur at the same position as the selected word, are generated by using the Recurrent Neural Network Language Model. We then introduce and explore how to use several linguistic resources as filters to select the best replacement words. In the experiments, we use the training data and the out-of-domain testing data of the CoNLL 2009 shared task. Training a SRL system from Lund University on the expanded training data gives us significant improvements in recall and F1 scores without penalizing precision scores for the selected roles over training the SRL on the original training data.

In summary the contributions of this paper are the following. First, we propose and evaluate a novel method for adapting a semantic role labelling system to a domain that is different from the one it is trained on. Second, we compare and evaluate several linguistic filters for selecting training examples adapted to the new domain. Finally, we demonstrate the value of language modelling information in the form of n-gram probabilities obtained from a large corpus, show how to integrate it in the semantic role labelling model and compare its results with the use of word embeddings and Brown word clusters.

II. Related work

SEMI-SUPERVISED approaches to semantic role labeling recently have received the attention of the computational linguistics community. Information obtained from a large collection of unlabelled texts have been used as extra features to improve the performance of SRL. Deep learning techniques based on semi-supervised embeddings have been used to improve a SRL system [5]. This track has been pursued further, using a deep neural network architecture to obtain good word representations in the form of word-embeddings [6]. The neural network technology is able to discover hidden representations of a word based on knowledge from its surrounding words possibly from the full sentence or discourse context using the large collection of unlabeled data. The hidden representations can take the form of predictive language models [7], that predict the next word given a n-gram of words, or in the form of word embeddings, the latter referring to a vector representation of a word that captures knowledge of its context [8]. A number of language models with hidden layers have been developed based on generative probabilistic approaches and applied to SRL. [2], [9] define a latent words language model as a graphical model where at each word position in a text the distribution of exchangeable words are generated. The exchangeable words are used as extra features to improve the performance of SRL on the CoNLL 2008 dataset especially when few training data are given to the learner. Recently, a novel technique for semantic frame identification has been introduced that uses distributed representations of predicates, achieving state-of-the-art results on FrameNet-style frame semantic analysis and strong results on PropBank-style semantic role labeling [10].

Besides the semi-supervised approaches that extend the feature set of SRL, there are other attempts to generate new training examples automatically by using unlabeled data. Lexical and syntactic similarity between labeled and unlabeled sentences have been considered as a graph alignment problem when generating training data [11]. More specifically, they represent sentences as dependency graphs and seek an optimal (structural) alignment between them. The language model of [2] has been used to generate new training examples by replacing the headword of temporal expression training examples in the task of temporal expression recognition [12].

In self-training, the existing model first labels unlabeled data. The newly labeled data is then treated as truth and combined with the actual labeled data to train a new model. This method has been effectively applied in syntactic parsing where an improvement of 1.1% over the previous best result for Wall Street Journal parsing has been reported [13]. There are several attempts of using self-training methods in semantic role labeling, but the gains are limited [14], [15]. Instead of using the totally new texts as training data, we only replace words in the manually labeled training instances in the hope to reduce the number of noisy training examples.

Recently, there have been several attempts to avoid the need for high-resource annotations (syntactic annotation, lemma etc.) when performing SRL. [16] couples latent syntactic representations, constrained to form valid dependency graphs or constituency parses, with the prediction task via specialized factors in a Markov random field. At both training and test time they marginalize over this hidden structure, learning the optimal latent representations for the problem. [17] compared various approaches for low-resource semantic role labeling at the state-of-the-art level and find that prior work in the low-resource setting can be outperformed by coupling the selection of feature templates based on information gain with a joint model that marginalizes over latent syntax.

There are also unsupervised attempts at semantic role labeling [18], [19], [20] which are easy to adapt to different domains, but recognition performance is usually much lower.

None of the above works consider both structural similarity and neutral language models as a source of evidence for generating training examples, nor do they evaluate different

TABLE I. MAIN PROPBANK SEMANTIC ROLES

Role	Description
A0	Agent - extern argument
A1	Patient/Theme - intern argument
A2	Indirect object / beneficiary / instrument / attribute / end state
AM-TMP	Temporal marker (when?)
AM-LOC	Location (where?)
AM-DIR	Direction
AM-MNR	Manner

approaches to similarity depending on the roles sought.

Instead of using a neutral language model, one may consider using other distributional semantic models as [21], [22], [23] to generate similar words in context. In our task, we compare the use of a neural language model and the Brown word clusters [23] which is one of the most commonly used clustering methods for semi-supervised learning. Brown word clustering is an agglomerative, bottom-up form of clustering that groups words into a binary tree of classes. The algorithm starts with each word in its own cluster. As long as there are at least two clusters left, the algorithm merges the two clusters that maximizes the quality of the resulting clustering. The quality of a clustering is viewed in the context of a class-based bigram language model. Given a clustering C that maps each word to a cluster, the class-based language model assigns a probability to the input text, where the maximum-likelihood estimate of the model parameters (estimated with empirical counts) are used. The the quality of the clustering C is defined as the logarithm of this probability [24]. The Brown word clusters can be used as candidates for our training example generating task as an alternative to the candidates proposed by a language model. However, unlike our proposed language model, the Brown clusters do not provide similarity scores of word pairs which can be used to rank candidates.

III. SEMANTIC ROLE LABELING

IN this section, we introduce the linguistic resources that we integrate in our SRL model and the standard SRL system that we use in our experiments.

A. Linguistic resources

1) PropBank: The Penn Proposition Bank (PropBank) [25] provides a corpus annotated with semantic roles, including participants appearing as arguments or adjuncts. The semantic roles defined in PropBank are quite generic and theory neutral (see Table I). A semantic frame which is evoked by a verb is represented as a role set. Each verb has several role sets corresponding to its possible senses. For example, in the sentence "Mary gave Peter a book at school yesterday", the role set $give.01$ can be annotated as: "[Mary *A0*] gave (*give.01*) [Peter *A2*] [a book *A1*] [at school *AM-LOC*] [yesterday *AM-TMP*]".

2) Nombank: The NYU NomBank project [26] can be considered part of the larger PropBank effort and is designed to provide argument structure for instances of about 5000 common nouns in the Penn Treebank II corpus. PropBank argument types and related verb frames files are used to provide a commonality of annotation. This enables the development of systems that can recognize regularizations of lexically and syntactically related sentence structures, whether they occur as verb phrases or noun phrases. The annotation of semantic frames in NomBank is similar to the annotation in PropBank.

3) VerbNet: In VerbNet [27], English verbs are grouped into different classes, adapting the previous verbal classification of [28]. Each verbal class takes different thematic roles and certain syntactic constraints that describe their superficial behavior. The semantic roles in VerbNet are more thematic than the ones in PropBank ($Agent$, $Patient$ instead of $A0$, $A1$). Members of a class share the same syntactic patterns with corresponding thematic roles. For example, two verbs "give" and "sell" (class $give-13.1.1$) in the two sentences "Mary gave Peter a book for 20 EUR" and "Mary sold Peter a book for 20 EUR" with the same syntactic pattern, should evoke two semantic frames with the same semantic role patterns as follows:
"[Mary *Agent*] gave (*give.01*) [Peter *Recipient*] [a book *Theme*] [for 20 EUR *Asset*]"
"[Mary *Agent*] sold (*sell.01*) [Peter *Recipient*] [a book *Theme*] [for 20 EUR *Asset*]"

4) SemLink: SemLink[4] is a project whose aim is to link together different lexical resources via a set of mappings. These mappings will make it possible to combine the different information provided by these different lexical resources for tasks such as inferencing. The mapping between VerbNet and PropBank is available in SemLink. Each frame in PropBank is linked to a suitable VerbNet class and each role label in the PropBank frame is mapped to a VerbNet role label. Since NomBank nouns also may reference PropBank verb senses, we are able to map NomBank frames to VerbNet classes by using SemLink and the references between NomBank and PropBank.

5) WordNet: WordNet [29] is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. Each of WordNet's 117000 synsets is linked to other synsets by means of a small number of "conceptual" relations. The main relation among words in WordNet is synonymy, as between the words "shut" and "close" or "car" and "automobile".

B. SRL system

SRL can be treated in the general framework of a classification task. It is often modeled in two stages: *predicate labeling* in which predicates are identified and disambiguated, and *argument labeling* in which arguments are identified and labeled.

In this paper, we use the open SRL software released by Lund University [30][5], a well-known SRL system, as it supports re-training of the model on new datasets, which is critical to our domain adaptation methodology. The system consists of a pipeline of independent, local classifiers that identify the predicate sense, the arguments of the predicates, and the argument labels. 32 features are used for argument

[4]http://verbs.colorado.edu/semlink/

[5]https://code.google.com/p/mate-tools/, version 2013

TABLE II. DENOTATION OF THE SYMBOLS USED IN THIS PAPER.

Symbol	Meaning
$\mathbf{S}_l$	Set of manually annotated sentences
$\mathbf{S}_t$	Testing set
$\mathbf{S}_{ul}$	Set of unlabeled sentences used to train the language model
$\mathbf{S}_u$	Set of unlabeled sentences generated automatically
$\mathbf{S}_{nl}$	Set of automatically annotated semantic frames of $\mathbf{S}_u$
$\mathbf{S}_{temp}$	Set of tuples of (sentence, word to be replaced, list of replacement words)
$\mathbf{V}$	Vocabulary of $\mathbf{S}_t$
N	Maximum number of replacement words for replacement candidate
z	Context window used to calculate replacement score

identification and classification including word form, lemma, POS, syntactic information of the predicate, the arguments, children of the arguments, left and right neighbours, etc. All the classifiers use the L2-regularized linear logistic regression from the LIBLINEAR package [31][6].

IV. OBJECTIVES AND TASK DEFINITION

IN this paper we discuss the problem of open domain SRL when the systems are applied on "target" domains other than the "source" domains they are trained on. Our approach is to automatically create new training examples that are "closer" to the target domain.

We start from manually annotated sentences and replace their most important words (predicate or argument) with words from the target domain. (This reduces the noise in the generated data over using entirely new sentences as typical in self-training methodologies.) To generate high quality training examples, the selected source-domain word and the replacement target-domain words must share "similar" or exchangeable syntactic structures and cluster the linguistic phrases that form a specific semantic role.

A language model (e.g., [32], [9]) gives us valuable information on both frequent and infrequent legitimate linguistic expressions, but we need additional mechanisms to constrain these expressions to allow the model to learn in the most efficient and effective way. We assume that language models combined with the information of linguistic expressions give us exchangeable words in context. The replacement words are considered as a cluster of words forming a specific semantic role.

In this respect, the goals of this paper are to:

- Set up a methodology for choosing unlabeled examples, guessing their labels, then using them as additional training data to improve the performance of a semantic role labeler on specific roles.
- Evaluate the methodology when the learned model is tested on texts that are out-of-domain compared to the texts on which they are trained.
- Critically discuss the gained performance and provide ideas for future semantic parsers that are trained on labeled and unlabeled examples.

Table II gives the notation for symbols used in this paper. Our goal is to learn a model that assigns semantic roles to the set of semantic frames of sentences in a test set. The learning takes into account a set of manually annotated sentences, a set of unannotated sentences, a language model and some linguistic resources.

In SRL, a sentence may contain more than one frame. Each semantic frame consists of one predicate that evokes the frame and several arguments that play a role in the frame. Predicates and arguments may be composed of more than one word. In this paper, we use *headword labeling*, which means if an argument consists of more than one word then the semantic role is assigned to only the headword. For instance, if "in the park" is the argument playing the role AM-LOC, then only the headword of the phrase "in the park", "in", is labeled with the label AM-LOC. Given a sentence s composed of n words $w_1, w_2, ..., w_n$, if the word at position p evokes a semantic frame f_p, then each word w_i in s will receive a label $r_{p,i} \in \mathbf{R} \cup \{NULL\}$ during the manual annotation for training and evaluation, where $\mathbf{R}$ is a set of predefined semantic roles and *NULL* means the empty label. If $r_{p,i} \neq NULL$, then w_i is the head of an argument of f_p with $r_{p,i}$ as the semantic role. In the approach that we describe in this paper, $\mathbf{R}$ is the set of PropBank/NomBank semantic roles (see Table I).

Given a set of manually annotated sentences $\mathbf{S}_l$, and a test set $\mathbf{S}_t$, a supervised SRL is trained on $\mathbf{S}_l$ and then annotates sentences in $\mathbf{S}_t$. In our approach, we use $\mathbf{S}_l$, the vocabulary $\mathbf{V}$ of the testing domain, a large set of unannotated sentences $\mathbf{S}_{ul}$, a language model L, and some linguistic resources, to generate a set of unannotated sentences $\mathbf{S}_u$ so that the semantic labels of the sentences in $\mathbf{S}_u$ can be guessed automatically. After guessing the semantic labels, $\mathbf{S}_{nl}$, the set of newly annotated semantic frames in $\mathbf{S}_u$, can be combined with the semantic frames of $\mathbf{S}_l$ to train the semantic role labeler, that is then evaluated on $\mathbf{S}_t$. Note that $\mathbf{S}_{nl}$ might contain some wrong labels. But, we hope that the proposed filters are able to eliminate noise in the automatically acquired examples.

In this paper, our focus is on how to automatically generate $\mathbf{S}_u$ based on a background language model and how to guess correctly the semantic labels of its sentences. The performance of the semantic role labeler that is trained on $\mathbf{S}_l \cup \mathbf{S}_{nl}$ is compared with a semantic role labeler that is only trained on $\mathbf{S}_l$ when labeling $\mathbf{S}_t$. To understand the meaning of an event, circumstance roles like Location, Time, Manner, Direction etc. are very important. However, the performance of semantic role labeling systems on those roles is often much lower than on the main roles as A0 or A1. In this research, we focus on improving SRL on the four main circumstance roles AM-LOC, AM-TMP, AM-MNR and AM-DIR.

V. METHODOLOGY

IN this section, we present a methodology to tackle semi-supervised semantic role labeling in an out-of-domain testing scheme. Our main idea is to replace important words from semantic frames in the training set by words in the vocabulary of the testing domain to create new semantic frames which are closer to the testing set. The new semantic frames can be used together with the original training set for training a SRL system. The steps of our methodology to generate new training examples and to train a SRL system are shown in Figure 1. Firstly, L is trained on $\mathbf{S}_{ul}$. Then, L, $\mathbf{S}_l$, $\mathbf{V}$ and the linguistic

[6]http://www.csie.ntu.edu.tw/~cjlin/liblinear/

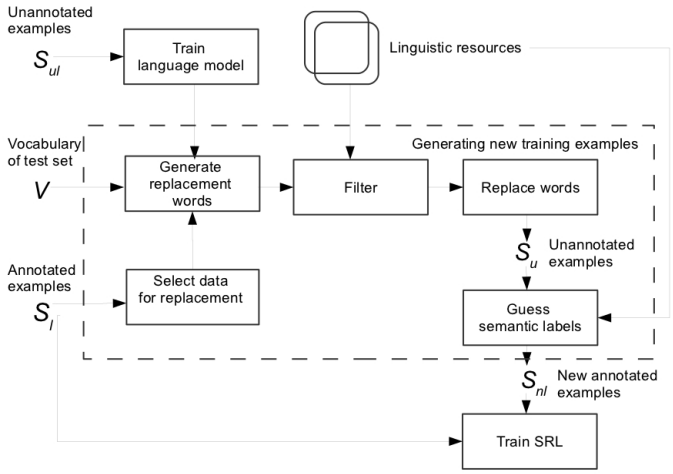

Fig. 1. Overview of the methodology to generate new training instances and train a SRL system.

resources are used to generate new training examples $\mathbf{S}_{nl}$. The detail of the algorithm to generate new training examples is presented below. After $\mathbf{S}_{nl}$ is generated, the SRL system is trained on $\mathbf{S}_l \cup \mathbf{S}_{nl}$.

A. General model for generating new training instances

In the dashed rectangle of Figure 1 it is shown how new training examples are generated. This process consists of five main steps:

Step 1. Selecting data for replacement. Since we focus on improving the performance of SRL on the four circumstance roles including AM-LOC, AM-TMP, AM-MNR and AM-DIR, and the majority of those roles are prepositional or subordinating conjunction phrases (PP) which have the "IN" tag in the Penn TreeBank [33] (see Table XI for the percentage of PP phrases per role in the CoNLL 2009 training data), we choose PP as the replacement target: *the sentences in S_l that have at least one frame with a PP argument are selected for the replacement*. Since the number of such sentences is much smaller than the total number of training examples, this keeps the computing cost from exploding. For each selected sentence, *the predicate or the objects of the PP arguments are selected to be replaced*. For example, given a sentence "Mary gave a book to Peter at school", the semantic frame "giving" has "gave" as predicate, and "at school" which is a PP as AM-LOC argument (only "at" is labeled with AM-LOC label), "gave" and "school" (the object of "at school") are selected as the candidates of the replacement.

Step 2. Generating replacement words for the selected words. A statistical language model assigns a probability to a sequence of m words by means of a probability distribution. For each word selected to be replaced, we use the language model L trained on $\mathbf{S}_{ul}$, and the vocabulary $\mathbf{V}$ of the testing domain, to generate a list of replacement words. Given a sentence composed of $w_1, w_2, ..., w_n$ where w_i is the word to be replaced, for each $nw_j \in \mathbf{V}$, the score of replacing w_i by nw_j is calculated by the probability of the sequence of words $w_{i-z}, w_{i-z+1}, ..., w_{i-1}, nw_j, w_{i+1}, w_{i+2}, ..., w_{i+z}$ obtained by putting nw_j in the context of w_i where z is size of the context window:

$$ReplacementScore(w_i, nw_j) = P(w_{i-z}, w_{i-z+1}, ..., w_{i-1}, nw_j, w_{i+1}, w_{i+2}, ..., w_{i+z})$$

This probability score is calculated by the language model. It is used to rank the replacement words in our algorithm. Since the size of $\mathbf{V}$ may be very large and the words at the end of the list may have a very low score which often represents noise, only N words that have the highest scores are chosen. After this step, we receive a ranked list of the top N replacement words for each replacement candidate.

Step 3. Applying filters to reduce noise in the list of replacement words. There may be a great deal of noise in the replacement words suggested by the language model since it does not take into account enough information (syntactic, semantic information etc.) to generate a replacement word that can be replaced perfectly for a word in a given sentence assuring the same semantic role. Thus, some linguistic filters are needed to improve the correctness and meaningfulness of the replacement.

Step 4. Replacing words in each sentence selected to be replaced by their replacement words that passed the filters, then we form a new unannotated set of sentences $\mathbf{S}_u$.

Step 5. Guessing semantic frames and their semantic labels for each sentence in $\mathbf{S}_u$ to have an annotated semantic frame set $\mathbf{S}_{nl}$.

In the following sections, we will present in more detail the language model used, some proposed filters, how to perform replacement and how to guess semantic role labels for the new sentences obtained by the replacement.

B. Language model

In this paper, we use the Recurrent Neural Network Language Model[7] (RNNLM) [32] [34], which is one of the most successful techniques for statistical language modeling. By using recurrent connections, these networks allow information (e.g., words from previous sentences in a discourse) to cycle inside and have an influence on the final language model obtained. The architecture of the RNNLM is shown in Figure 2. The input layer consists of a vector $\mathbf{w}(t)$ that represents the current word w_t encoded as 1 of V (V is the vocabulary), and of vector $\mathbf{s}(t-1)$ that represents output values in the hidden layer from the previous time step. After the network is trained, the output layer $\mathbf{y}(t)$ represents $P(w_{t+1}|w_t, \mathbf{s}(t-1))$. The network is represented by input, hidden and output layers and corresponding weight matrices - matrices $\mathbf{U}$ and $\mathbf{W}$ between the input and the hidden layer, and matrix $\mathbf{V}$ between the hidden and the output layer. Output values in the layers are computed as follows:

$$\mathbf{s}(t) = f(\mathbf{U}\mathbf{w}(t) + \mathbf{W}\mathbf{s}(t-1)) \quad (1)$$

$$\mathbf{y}(t) = g(\mathbf{V}\mathbf{s}(t)) \quad (2)$$

[7] http://www.fit.vutbr.cz/ imikolov/rnnlm/

Algorithm 1 Generate novel training examples.

procedure GENERATENEWEXAMPLE(L, $\mathbf{S}_l$, $\mathbf{V}$, z, N)
  $\mathbf{S}_u = \emptyset$, $\mathbf{S}_{nl} = \emptyset$, $\mathbf{S}_{temp} = \emptyset$;
  for each replacement candidate sentence $s \in \mathbf{S}_l$ **do**
    for each replacement candidate word w_i in s **do**
      for each $nw_j \in \mathbf{V}$ **do**
        $ReplacementScore(w_i, nw_j) = P(w_{i-z}, w_{i-z+1}, ..., w_{i-1}, nw_j, w_{i+1}, w_{i+2}, ..., w_{i+z})$ obtained by using L;
      end for
      $\mathbf{List}_i$ = Top N of nw_j with highest $ReplacementScore$
      $\mathbf{S}_{temp} = \mathbf{S}_{temp} \cup (s, w_i, \mathbf{List}_i)$;
    end for
  end for
  for each replacement candidate sentence $s \in \mathbf{S}_l$ **do**
    %%% Predicate replacement %%%
    for each replacement candidate predicate w_p in s **do**
      $\mathbf{List}_p$ = Top N of replacement word nw_j of w_p stored in $\mathbf{S}_{temp}$
      for each replacement word nw_j in $\mathbf{List}_p$ **do**
        if nw_j passes **filters** **then**
          s' = the sentence obtained by replacing w_p by nw_j in s;
          $\mathbf{S}_u = \mathbf{S}_u \cup s'$;
          f'= the semantic frame evoked by nw_j in s';
          Guess_semantic_role_labels_for_Predicate_Replacement(f');
          $\mathbf{S}_{nl} = \mathbf{S}_{nl} \cup f'$;
        end if
      end for
    end for
    %%% Argument replacement %%%
    for each replacement candidate argument w_i in s **do**
      $\mathbf{List}_i$ = Top N of replacement word nw_j of w_i stored in $\mathbf{S}_{temp}$
      for each replacement word nw_j in $\mathbf{List}_i$ **do**
        if nw_j passes **filters** **then**
          s' = the sentence obtained by replacing w_i by nw_j in s;
          $\mathbf{S}_u = \mathbf{S}_u \cup s'$;
          f'= the semantic frame in s' evoked by the word that is the predicate of f;
          Guess_semantic_role_labels_for_Argument_Replacement(f') ;
          $\mathbf{S}_{nl} = \mathbf{S}_{nl} \cup f'$;
        end if
      end for
    end for
  end for
  Return $\mathbf{S}_{nl}$
end procedure

where *f(z)* and *g(z)* are sigmoid and softmax activation functions, respectively. The model is trained using the back-propagation algorithm to maximize the data conditional likelihood:

$$\prod_t P(\mathbf{y}(t)|w(1)w(2)...w(t)) \tag{3}$$

The output layer y represents a probability distribution of the next word w_{t+1} given the history:

$$\mathbf{y}^*(t) = \arg\max P(y(t)|w(1)w(2)...w(t)) \tag{4}$$

We call the vectors in the matrix between the input and the hidden layer word vector *word embeddings* (also known as word vectors). Each word is associated with a real valued vector in the K-dimensional output space of the RNNLM.

Using these distributed representations of words, one can predict the next word given the previous n-1 words to form a n-gram language model.

C. Filters

Because the list of top N replaceable words returned by the language model may contain a great deal of noise, we propose specific filters to improve the performance of the system.

1) Part-Of-Speech filter (POS filter): We keep replacement word nw_j for w_i if nw_j has the same POS tag as w_i, when replacing w_i in sentence s.

2) WordNet filter: We keep replacement word nw_j for w_i if nw_j and w_i are connected by a semantic relation in WordNet: synonym, hypernym, co-hypernym (sharing a hypernym), hyponym, and meronym.

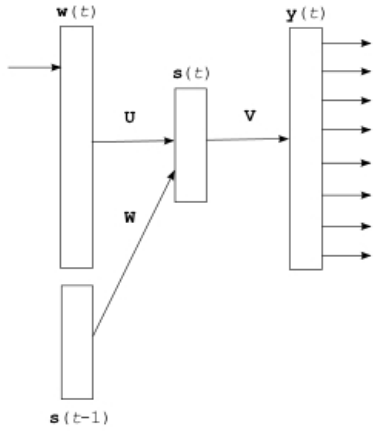

Fig. 2. Simple recurrent neural network.

TABLE III. ADVANTAGES AND DISADVANTAGES OF THE PROPOSED FILTERS

Filter	Advantages	Disadvantages
POS	Improves the syntactic quality of the new training examples, therefore keeps precision from dropping. The number of new training examples may be very high therefore it can improve recall in some cases.	The very large number of new training examples may also increase the noise in the new training example, so it may reduce the precision in some cases.
WordNet	Improves the semantic quality of the new training examples, therefore keeps precision from dropping.	Number of new training examples might be limited.
Predicate	Improves both the semantic and syntactic quality of the new training examples, therefore keeps precision from dropping.	Number of new training examples might be limited.

3) Predicate filter: We keep replacement predicate word nw_j for a predicate w_p that evokes a frame f_p if f_p and one frame evoked by nw_j are mapped to the same VerbNet class and the mappings from those two frames to the VerbNet class are defined in SemLink (see Section III-A4).

The advantages and disadvantages of all the filters are discussed in Table III.

D. Replacing words and guessing semantic labels

A sentence s composed of n words $w_1, w_2, ..., w_n$ is a replacement candidate. In what follows, we present how to replace a predicate and an argument in s to obtain new training instances.

1) Predicate replacement: The predicate w_p evoking semantic frame f_p in s is a replacement candidate. $List_p$ is the list of replacement words of w_p. For each $nw_j \in List_p$ that passed the filtering step, we replace w_p by nw_j in sentence s and obtain sentence s' composed of n words $w_1, w_2, ...w_{p-1}, nw_j, w_{p+1}, ..., w_n$. If nw_j has passed Predicate filter – which we use as a default filter in this setting – the argument structure of the frame evoked by nw_j is similar to the argument structure of the frame evoked by w_p: nw_j also invokes a semantic frame f'_p in s'. In order to predict

TABLE IV. ROLE MAPPING OF "SIMPLE_DRESSING-41.3.1" LINKED TO BOTH "WEAR.01" AND "DON.01"

Role of *simple_dressing-41.3.1*	Role of *wear.01*	Role of *don.01*
Agent	Arg0	Arg0
Theme	Arg1	Arg1

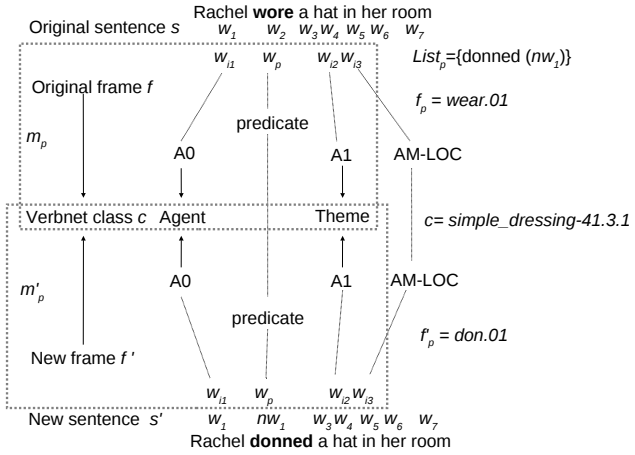

Fig. 3. An example of predicate replacement.

the sense and role labels of the new semantic frame, we use the mappings between PropBank/NomBank semantic frames and VerbNet classes that can be found in SemLink: f'_p is the semantic frame of nw_j so that both f_p and f'_p are mapped to a same VerbNet class c. We call the mappings from f_p and f'_p to c, m_p and m'_p, respectively. Each argument w_i of f_p (role $r_{p,i}$) is also an argument of f'_p (role $r'_{p,i}$). If $r_{p,i}$ is a circumstance role AM-s, then $r'_{p,i} = r_{p,i}$, else $r'_{p,i} = {m'_p}^{-1}(m_p(r_{p,i}))$. For example, the sentence "Rachel wore a hat in her room" has the frame "wear.01" (wore) with "Rachel" as A0, "hat" as A1, "in" (the head of the prepostion phrase "in her room") as AM-LOC, and the predicate "wore" has "donned" as a replacement word. By replacing "wore" by "donned" in the sentence, we have a new sentence "Rachel donned a hat in her room" and "donned" evokes a new frame. In SemLink, we can find the VerbNet class "simple_dressing-41.3.1" linked to the Propbank frame "wear.01" and one PropBank frame of the predicate "don", "don.01". The role mapping between the VerbNet class and the two frames can be found in Table IV. By applying our method, we have a new frame "don.01" with "Rachel" as A0 (mapped to the "Agent" VerbNet role), "hat" as A1 (mapped to the "Theme" VerbNet role), and "in" as AM-LOC (circumstance role) (See Figure 3).

2) Argument replacement: We perform replacement on arguments that are prepositional or subordinating phrases such as "on the table" or "before next morning". The object of a prepositional or subordinating phrase as "table" or "morning" can be replaced by its replacement words. We consider that w_i is an argument of the semantic frame f_p evoked by w_p in s. If w_i is a preposition ("in", "on", etc.) or a subordinating conjunction ("after", "until" etc.) and w_o is the object of the phrase headed by w_i, then w_o is a candidate for the

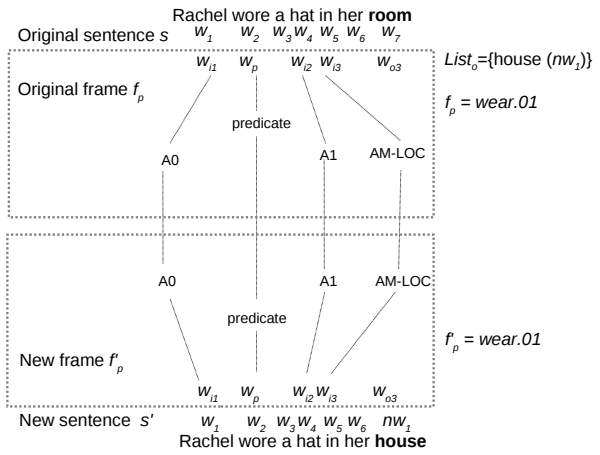

Fig. 4. An example of argument replacement.

TABLE V. NUMBER OF TRAINING/TESTING INSTANCES PER ROLE (CONLL 2009 DATASETS).

Data	A0	A1	A2	AM-LOC	AM-TMP	AM-MNR	AM-DIR
$\mathbf{S}_l$	99388	146548	46741	10387	23347	11837	1146
$\mathbf{S}_t$	741	1004	227	87	121	131	52

TABLE VI. SYMBOLS THAT DENOTE EXPERIMENT SETTINGS

Symbol	Meaning
WPred-PR	WordNet filter, Predicate filter and predicate replacement.
PWPred-PR	Part-Of-Speech filter, Predicate filter, WordNet filter, and predicate replacement.
PPred-PR	Part-Of-Speech filter, Predicate filter and predicate replacement.
W-AR	WordNet filter, argument replacement.
PW-AR	Part-Of-Speech filter, WordNet filter, argument replacement.
P-AR	Part-Of-Speech filter, argument replacement.
WE	Using Word Embeddings as extra features.
BPred-PR	Using Brown word classes as candidates, Predicate filter and predicate replacement (no Language Model)
LBPred-PR	Using Brown word classes as filter, Predicate filter and predicate replacement

replacement. $List_o$ is the list of replacement words of w_o. For each $nw_j \in List_o$ that passed the filtering step, we replace w_o by nw_j in s and obtain the new sentence s' composed of n words $w_1, w_2, ...w_{o-1}, nw_j, w_{o+1}, ..., w_n$. In s', w_p also invokes a frame $f'_p = f_p$ and its arguments and roles are the same as f_p. That means if w_i is an argument of f_p with the role $r_{p,i}$, then it is also an argument of f'_p with the role $r'_{p,i} = r_{p,i}$. For example, in "Rachel wore a hat in her room", "in" (the head of the prepositional phrase "in her room") is assigned the label of AM-LOC. "room" is the object of the prepositional phrase "in her room" and it has "house" as a replacement word. After performing argument replacement, we have a new sentence "Rachel wore a hat in her house". In this new sentence, "wore" evokes a new semantic frame "wear.01" with "Rachel", "hat" and "in" as A0, A1, AM-LOC respectively which are the same as the roles in the original frame. But in this new semantic frame, instead of "room", "house" is the object of the PP headed by "in" (See Figure 4).

VI. EXPERIMENTS

A. Experimental setup

In this section we describe our experiments to evaluate the performance of our method on the portability of the semantic role labeling system. Our experiments are targeted to answer the following questions:

- Does our methodology improve performance on the targeted roles: AM-LOC, AM-TMP, AM-MNR, and AM-DIR?
- How effective is the language model for selecting replacement words?
- Is predicate replacement or argument replacement better?
- Which of the filters are most helpful?

As specified in Section IV, in our experiments, we apply our methodology to improve the classification results of the four semantic roles: AM-LOC, AM-TMP, AM-MNR, AM-DIR. We use an open semantic parser from Lund University [30] (in the default mode, using gold predicate identification), which allows to re-train the model easily. The fully manually annotated CoNLL 2009[8] training data (in English, parts of the Wall Street Journal corpus[9]) and the out-of-domain test set of CoNLL 2009 (in English, the fiction part from the Brown corpus[10]) are used as the annotated training set $\mathbf{S}_l$ and the testing set $\mathbf{S}_t$, respectively. The information on $\mathbf{S}_l$ and $\mathbf{S}_t$ is given in Table V. The RNNLM[11] is trained on the corpus containing the first 80 million words of the Reuters corpus and 0.4 million words of the Brown corpus. In our experiment, the number of hidden units is set to 300, number of epochs is 13. By using the language model, we generate a list of the top 400 replacement words for each replacement candidate ($N = 400$). The context window size used in our experiments, z, equals 5. After applying several combinations of filters, we replace each replacement candidate by its replacement words. We then guess the semantic labels of the new semantic frames, and obtain a new annotated semantic frame set $\mathbf{S}_{nl}$. To evaluate our methodology, we train Lund SRL on $\mathbf{S}_l \cup \mathbf{S}_{nl}$, then use the trained model to label the arguments of semantic frames in $\mathbf{S}_t$. This result is compared with the result obtained by using the SRL model trained on only $\mathbf{S}_l$ to label the arguments of semantic frames in $\mathbf{S}_t$.

In another experiment, we perform a comparison to using word embedding as extra features. The word embeddings obtained when training the language model are used as extra features for SRL system. The system is trained on the original training data $\mathbf{S}_l$ and then tested on $\mathbf{S}_t$ as in the other settings.

Experiment settings are represented by codes denoting the filters, and the replacement method used (see Table VI).

B. Experimental results and discussions

The baseline obtained by training the SRL system on all semantic frames of S_l and testing on S_t is described in Table

[8]http://ufal.mff.cuni.cz/conll2009-st/

[9]http://catalog.ldc.upenn.edu/LDC2012T04

[10]http://catalog.ldc.upenn.edu/LDC2012T04

[11]http://rnnlm.org/

TABLE VII. RESULTS IN TERMS OF RECALL, PRECISION AND F1 PER ROLE WHEN APPLYING THE BASELINE SRL (%).

Role	Precision	Recall	F1
A0	78.37	75.30	76.81
A1	74.45	71.12	72.75
A2	54.98	51.10	52.97
AM-DIR	51.85	26.92	35.44
AM-LOC	43.59	39.08	41.21
AM-MNR	45.97	43.51	44.71
AM-TMP	48.46	52.07	50.20

VII.

1) Performance on circumstance roles: The first row of Table VIII shows performance of the best overall model (WPred-PR; replacement words from RNN + WordNet and predicate filters + predicate replacement) on the circumstance roles AM-LOC, AM-TMP, AM-MNR and AM-DIR. Overall, we see an average F1 gain of +1.90 across the four roles, primarily driven by increases of recall for each of the roles, with smaller gains in precision (or in the case of AM-MNR in a few cases a slight drop). However, we also notice that different circumstantial roles demand different linguistic filters: the WPred-PR method yields an F1 gain of 3.36 and 2.27% for the AM-LOC and AM-TMP role recognition, respectively. The labelling of the AM-MNR role gains 2.84% with the P-AR method, while the AM-DIR role recognition improves even with 14% when using the PPred-PR method. These results confirm our hypothesis that creating new training examples targeting specific roles requires a different approach depending on the role. In fact, AM-LOC and AM-TMP have a large PP, and are characterized by a reasonable size of training instances in the CoNLL dataset, so WPred-PR returns enough new training examples. Meanwhile, for AM-DIR and AM-MNR, which has a lower percentage of PP and/or smaller size of training instances, we need the settings with only POS filter (PPred-PR, P-AR) to get more new training examples.

Though our focus is on the circumstance roles, we also show performance of the model on A0, A1 and A2 when they occur in prepositional phrases (Table IX).

When we consider our overall best performing method (WPred-PR), we see a small increase for A0 and small decreases for A1 and A2, but again different models behave differently for different roles. Our approach did not target improvements of A0, A1 and A2 roles. Moreover, we only have selected labeled examples that contain PP phrases to generate new examples. Table XI and Table XII show that the number of prepositional phrases (and therefore the number of new training instances) is very limited for the A0, A1 and A2 roles to learn a better SRL labelling from.

To further demonstrate that our method does not hurt the overall SRL performance for the prediction of all semantic roles, we compare the performance averaged over all roles and predicate disambiguation over the baseline using the CoNLL 2009 scorer of our best method (WPred-PE), and get the positive gain obtained of +0.46% (the result of the baseline is 72.39%). It shows that the performance improvement from our method on the less common circumstance roles outweighs any minor performance losses on the much more common main argument roles.

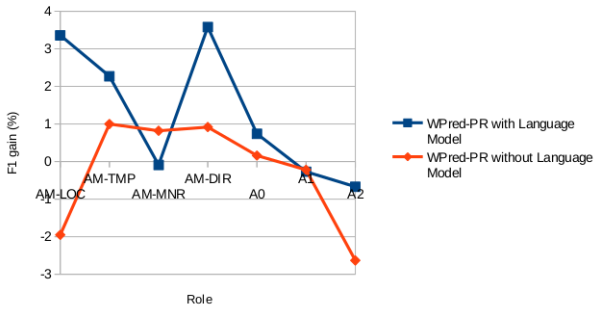

Fig. 5. F1 gains when using the language model (RNNLM).

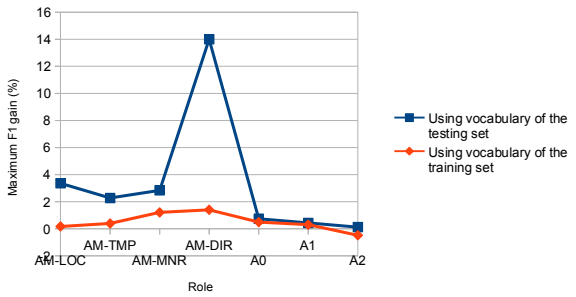

Fig. 6. Comparison of F1 gains using the vocabulary of the testing/training datasets.

2) Effectiveness of the language model: Figure 5 shows the performance gains over the baseline when using the language model (RNNLM) in word replacement compared to using the full vocabulary of the test domain in word replacement. Both models here use WordNet and predicate filters + predicate replacement (WPred-PE), but in the first we select the top N words predicted by the RNNLM, and in the second, we select all words from the vocabulary $\mathbf{V}$ of the test domain to be used for replacement. The former model yields a larger gain except for AM-MNR and A1 roles.

Note that the gain in performance primarily comes from the language model's ability to predict unseen words from the target domain. Figure 6 shows the performance when $\mathbf{V}$ is instantiated as the vocabulary of the training domain vs. the vocabulary of the testing domain. Performance gains are minimal in the former case, but large in the latter case.

Another use of language information discussed in Section II is to simply add the word embeddings (word vectors) as features in the SRL training and prediction. The word embeddings are obtained with the same recurrent neural network trained over the same data as our RNNLM. Table X compares this approach (WE) to our use of language models to generate new training instances. The results show the already beneficial effect of using the word embeddings, especially in terms of recall, but show that our best model outperforms the use of word embeddings in terms of gain in precision, yielding an overall better performance. This finding indirectly shows that the use of linguistic resources is beneficial to more precisely use the background knowledge learned by the recurrent neural

TABLE VIII. PERFORMANCE GAINS (%) FOR THE SEMANTIC ROLES AM-LOC, AM-TMP, AM-MNR AND AM-DIR.

Method	AM-LOC			AM-TMP			AM-MNR			AM-DIR			AM-
	R	P	F1	R	P	F1	R	P	F1	R	P	F1	F1
WPred-PR	**+5.75**	+0.73	**+3.36**	**+4.96**	**+0.13**	**+2.27**	+0.76	-1.01	-0.09	+3.85	+1.48	+3.58	**+1.90**
PWPred-PR	0.00	**+1.15**	+0.51	+3.31	-0.26	+1.34	0.00	0.00	0.00	+1.92	**+8.15**	+3.52	+1.00
PPred-PR	0.00	-5.39	-2.58	+3.31	-3.50	-0.57	+3.05	-1.44	+0.82	**+15.38**	+7.61	**+14.00**	+1.39
W-AR	-2.30	-8.42	-5.26	+2.48	-1.65	+0.18	+2.29	**+1.65**	+1.99	+5.77	+4.81	+6.02	+0.35
PW-AR	-2.30	-5.04	-3.57	+1.65	-2.03	-0.39	0.00	-0.37	-0.17	+1.92	+3.70	+2.53	-0.63
P-AR	-3.45	-11.96	-7.70	0.00	-9.81	-5.83	**+4.58**	+1.05	**+2.84**	+9.62	+5.72	+9.26	-1.33

TABLE IX. PERFORMANCE GAINS (%) FOR THE SEMANTIC ROLES A0, A1 AND A2.

Method	A0			A1			A2		
	R	P	F1	R	P	F1	R	P	F1
WPred-PR	**+1.35**	+0.08	**+0.74**	+0.50	-1.09	-0.27	-0.88	-0.43	-0.67
PWPred-PR	-0.13	**+0.64**	+0.23	+0.20	-0.18	+0.02	0.00	+0.26	**+0.12**
PPred-PR	-0.13	-1.22	-0.66	**+2.19**	-2.37	-0.05	-0.44	**+0.31**	-0.09
W-AR	+0.54	-0.21	+0.18	+0.30	-0.15	+0.08	**+0.44**	-1.31	-0.38
PW-AR	-0.13	-1.01	-0.56	+0.80	-2.90	-1.01	-0.44	-7.46	-3.93
P-AR	+0.13	+0.03	+0.08	+0.50	**+0.37**	**+0.43**	-0.44	-0.99	-0.70

TABLE X. PERFORMANCE GAINS (%) OF OUR BEST MODEL AND THE USES OF WORD EMBEDDINGS AND BROWN WORD CLUSTERS ACROSS THE FOUR TARGETED ROLES.

Method	R	P	F1
WPred-PR	+3.58	+0.06	+1.90
WE	+2.73	-0.21	+1.28
BPred-PR	+2.10	-1.25	+0.5
LBPred-PR	+1.53	-1.83	-0.06

network.

In Table X, we also compare the use of Brown word clusters in two predicate replacement settings to our best model: First, instead of using RNNLM, Brown word clusters are used as candidates for the replacement. That means, for each predicate in the training data selected for the replacement, we consider all words in the testing domain that are in the same class as the predicate as the replacement candidates (BPred-PR). Second, instead of using WordNet filter, Brown word clusters are used to filter the candidates suggested by the RNNLM (LBPred-PR). The Brown word clusters are trained by using the implementation of [24] on the same corpus used to train the RNNLM. The number of clusters is 500[12] . Table X shows the already beneficial effect of using Brown clustering in terms of recall, but show that in terms of gains in precision and F1, our best model outperforms both of the two settings using Brown clustering.

3) Comparing predicate vs. argument replacement: The six rows of Table VIII compare predicate and argument replacement, with the top three representing predicate replacement and the bottom three representing argument replacement. On the average, we see higher gains from predicate replacement than argument replacement, with the worst predicate replacement model (PWPred-PR) getting an average F1 gain of +1.00, and the best argument replacement model (W-AR) getting an average F1 gain of +0.35. One explanation for the success of predicate replacement over argument replacement may be the feature set used by the Lund SRL. Replacing the predicate in each semantic frame changes the "predicate word", "predicate lemma", and "the sense of predicate" (frame name) features [30]. Replacing prepositional/subordinating conjunction phrase arguments changes only the features "the children word set" and "children Part-Of-Speech set" which are only included for nominal predicates. This means that predicate replacement often passes more new information to the SRL system.

4) Effectiveness of different filter types: The top three rows of Table VIII compare systems with different filters enabled. These systems all use the predicate filter because it is required when using predicate replacement. Comparing the first and second row, we can see that adding the part-of-speech filter generally lowers performance (except for the case of AM-DIR precision). Comparing the second and third row, we can see that adding the WordNet filter generally improves performance (except for the case of AM-MNR and AM-DIR recall).

We also explored several variants of the WordNet filter using different semantic relation types (synonym, hypernym, co-hypernym, hyponym and meronym) as shown in Figure 7. Only synonym and co-hypernym relations gave consistent F1 gains for both predicate replacement and argument replacement (and were thus what was used in all other experiments in this paper that reference the WordNet filter).

[12]We also test with 250, 750 and 1000 clusters. 500 is the number gives us the best scores.

TABLE XI. PERCENTAGE OF PREPOSITIONAL/SUBORDINATING CONJUNCTION PHRASES PER ROLE IN $\mathbf{S}_l$

A0	A1	A2	AM-LOC	AM-TMP	AM-MNR	AM-DIR
6%	18%	27%	67%	40%	36%	26%

TABLE XII. SIZE OF THE NEW TRAINING EXAMPLES ($\mathbf{S}_{nl}$) PER ROLE (% OF THE ORIGINAL TRAINING EXAMPLES IN $\mathbf{S}_l$).

Data	A0	A1	A2	AM-LOC	AM-TMP	AM-MNR	AM-DIR
WPred-PR	8.76	8.73	5.11	23.89	23.59	15.46	41.24
PWPred-PR	1.01	1.33	0.87	3.23	4.35	2.39	3.75
PPred-PR	14.38	16.06	10.00	44.59	47.33	26.36	59.79
W-AR	3.16	9.85	14.12	46.10	59.27	9.66	19.91
PW-AR	1.09	4.58	5.40	20.80	36.63	5.10	11.56
P-AR	51.50	96.18	179.12	493.96	222.83	100.30	255.03

VII. FURTHER DISCUSSIONS AND FUTURE WORK

A. When does our methodology work?

The most important factor that affects the performance of our methodology is the number of new training examples that can be created. It depends on the relatedness between words in the vocabulary of the test set and the words selected to be replaced in the training data. If the words in the vocabulary of the test set have no relation to the context and the words of the training examples, then they cannot appear in the list of replacement

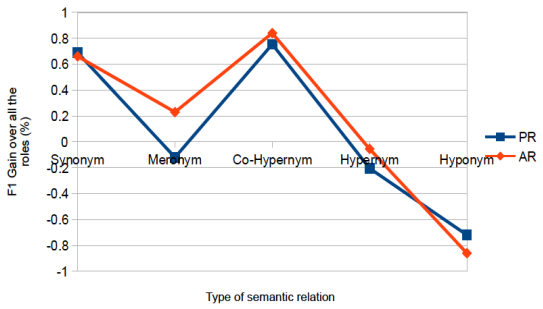

Fig. 7. Comparison of WordNet filters.

words suggested by the language model and cannot pass the filters, therefore we cannot create new training examples from those words. Moreover, the problem of a low percentage of prepositional/subordinating conjunction phrases in some roles may limit the number of new training examples, and therefore limit the performance of our system. For example, in the experiment with the setting WPred-PR, which gives us the best F1 gains on the circumstance roles, although the total instances of AM-LOC is smaller than that of AM-MNR in the original training data, the percentage of prepositional/subordinating conjunction phrases in AM-LOC is much higher than in AM-MNR, so the percentage of new training examples for AM-LOC is higher than that of AM-MNR, and the performance of our system when recognizing AM-LOC roles is better than when recognizing AM-MNR roles. Therefore, our methodology works better when the words from the vocabulary of the testing domain have some semantic/syntactic relations with the words in the training domain, and the percentage of prepositional/subordinating conjunction phrases in the training examples is not low.

B. Limitations

There are several other factors that limit the performance of our system. First of all, the errors of syntactic parsing (wrong objects of prepositional/subordinating conjunction phrases, wrong prepositional phrase attachment) may cause problems for our methodology. Replacing a wrong object of a prepositional/subordinating conjunction phrase will increase the number of noisy automatically generated training examples. Applying predicate replacement on a wrong prepositional phrase attachment will lead to a wrong new semantic frame. Secondly, the performance of our methodology also depends on the SRL model's feature list. For many existing semantic role labelers, replacing one word in a semantic frame changes only one or a few features while the other features are still the same, and it may not be able to help us to improve the results. Finally, the WordNet filter may return a wrong replacement word because of the word disambiguation problem. However the language model provides the context in which the word is used possibly simulating word sense disambiguation.

C. Future work

The reported research shows that through the use of a language model and several linguistic resources we can find replacement words, i.e. clusters of words in a given sentence that are exchangeable with regard to the targeted semantic role. In this way, we have leveraged the cluster hypothesis, which is a necessary condition for successful semi-supervised learning methods [35]. The proposed model has resulted in higher F1 measures in the recognition of the targeted semantic roles. In future work, we will further refine our models on the one hand by focusing on other semantic labeling tasks and on the other hand by improving machine learning models that integrate the language resources used in this paper. In this respect, the investigation of effective linguistic features for the targeted semantic labels seems valuable. This is in line with the above finding that creating new training examples targeting specific semantic roles requires a different approach depending on the role. Furthermore, it is very interesting to further investigate how the relation between the testing and training data affects the performance of the methodology and to further refine the proposed model so as to better understand the mechanisms of an effective domain transfer. Our methodology can be applied to other case studies in SRL by changing the replacement candidates toward the new goals. For example, one may improve the performance of SRL on movement semantic frames by performing predicate replacement on movement verbs.

VIII. Conclusions

In this paper, we propose a methodology that uses a language model and some linguistic resources to adapt SRL to another domain by replacing words in the training data so as to automatically create new training examples. Our simple but effective methodology can be applied to other problems to create new training examples. More specifically, we propose to select semantic frames that contain at least one preposition phrase argument as target for the replacement to improve the results of SRL on four important circumstance roles AM-LOC, AM-TMP, AM-MNR and AM-DIR. There are two methods of replacement: predicate replacement and argument replacement. The replacement words can be generated by using a language model. To reduce the noise in the list of replacement words, three linguistics filters including Part-Of-Speech filter, WordNet filter and Predicate filter, are proposed. In the experiments, we have shown that our method can give us promising recall and F1 gains without penalizing the precision score on the four circumstance roles targeted. For the four roles AM-LOC, AM-TMP, AM-MNR and AM-DIR, using the WordNet filter (filtering for synonym and co-hypernym) in combination with the Predicate filter with predicate replacement is the best method. We also prove the importance of using a language model. The easy combination of language modelling information with linguistic filters yielding results that outperform the uses of word embeddings and Brown word clusters makes this model a good choice for generating extra training examples from unlabelled data. The language model reduces noise and suggests the most probable

replacement candidates. Finally, we also show that using the vocabulary of the application domain is helpful in the transfer of a SRL model to a new domain.

ACKNOWLEDGMENT

This work is funded by the EU ICT FP7 FET project "Machine Understanding for interactive StorytElling" (MUSE) http://www.muse-project.eu/.

REFERENCES

[1] M. Palmer, D. Gildea, and N. Xue, *Semantic Role Labeling*, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2010.

[2] K. Deschacht and M.-F. Moens, "Semi-supervised semantic role labeling using the latent words language model," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2009*. ACL, 2009, pp. 21–29.

[3] F. Huang, A. Ahuja, D. Downey, Y. Yang, Y. Guo, and A. Yates, "Learning representations for weakly supervised natural language processing tasks," *Computational Linguistics*, vol. 40, pp. 85–120, 2013.

[4] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 238–247. [Online]. Available: http://www.aclweb.org/anthology/P14-1023

[5] J. Weston, F. Ratle, and R. Collobert, "Deep learning via semi-supervised embedding," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1168–1175. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390303

[6] R. Collobert, "Deep learning for efficient discriminative parsing," in *AISTATS*, 2011.

[7] W. De Mulder, S. Bethard, and M.-F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling," *Computer Speech and Language*, vol. 30, no. 1, pp. 61–98, 2015. [Online]. Available: https://lirias.kuleuven.be/handle/123456789/462643

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[9] K. Deschacht, J. De Belder, and M.-F. Moens, "The latent words language model," *Computer Speech and Language*, vol. 26, no. 5, pp. 384–409, Oct. 2012. [Online]. Available: https://lirias.kuleuven.be/handle/123456789/344914

[10] K. M. Hermann, D. Das, J. Weston, and K. Ganchev, "Semantic frame identification with distributed word representations," in *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics*, 2014.

[11] H. Fürstenau and M. Lapata, "Semi-supervised semantic role labeling via structural alignment," *Comput. Linguist.*, vol. 38, no. 1, pp. 135–171, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1162/COLI_a_00087

[12] O. Kolomiyets, S. Bethard, and M.-F. Moens, "Model-portability experiments for textual temporal analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, ser. HLT '11. Stroudsburg, PA, USA: ACL, 2011, pp. 271–276. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002736.2002793

[13] D. McClosky, E. Charniak, and M. Johnson, "Effective self-training for parsing," in *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, ser. HLT-NAACL '06. Stroudsburg, PA, USA: ACL, 2006, pp. 152–159. [Online]. Available: http://dx.doi.org/10.3115/1220835.1220855

[14] Y. Si, Z. Zhang, T. Li, J. Pan, and Y. Yan, "Enhanced word classing for recurrent neural network language model," in *JICS: Journal of Information and Computational Science, Vol. 10*, 2013, pp. 3595–3604.

[15] R. Samad Zadeh Kaljahi, "Adapting self-training for semantic role labeling," in *Proceedings of the ACL 2010 Student Research Workshop*. ACL, 2010, pp. 91–96. [Online]. Available: http://aclweb.org/anthology/P10-3016

[16] J. Naradowsky, S. Riedel, and D. A. Smith, "Improving nlp through marginalization of hidden syntactic structure," in *Proceedings of the Conference on Empirical methods in natural language processing, EMNLP '12*. Jeju, Korea: ACL, July 2012.

[17] M. Gormley, M. Mitchell, B. Van Durme, and M. Dredze, "Low-resource semantic role labeling," in *Association for Computational Linguistics (ACL)*, June 2014. [Online]. Available: http://www.cs.jhu.edu/~mrg/publications/srl-acl-2014.pdf

[18] R. S. Swier and S. Stevenson, "Unsupervised semantic role labelling," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2004*. ACL, 2004, pp. 95–102.

[19] O. Abend, R. Reichart, and A. Rappoport, "Unsupervised argument identification for semantic role labeling," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ser. ACL '09. Stroudsburg, PA, USA: ACL, 2009, pp. 28–36. [Online]. Available: http://dl.acm.org/citation.cfm?id=1687878.1687884

[20] T. Grenager and C. D. Manning, "Unsupervised discovery of a statistical verb lexicon," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2006*, D. Jurafsky and É. Gaussier, Eds. ACL, 2006, pp. 1–8.

[21] M. Baroni and A. Lenci, "Distributional Memory: A general framework for corpus-based semantics," *Computational Linguistics*, vol. 36, no. 4, pp. 673–721, 2010.

[22] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Int. Res.*, vol. 37, no. 1, pp. 141–188, Jan. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1861751.1861756

[23] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Comput. Linguist.*, vol. 18, no. 4, pp. 467–479, Dec. 1992. [Online]. Available: http://dl.acm.org/citation.cfm?id=176313.176316

[24] P. Liang, "Semi-supervised learning for natural language," in *Master Thesis, MIT*, 2005.

[25] M. Palmer, P. Kingsbury, and D. Gildea, "The proposition bank: An annotated corpus of semantic roles," *Computational Linguistics*, vol. 31, no. 1, pp. 71–106, 2005.

[26] A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman, "The nombank project: An interim report," in *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, A. Meyers, Ed. Boston, Massachusetts, USA: ACL, May 2 - May 7 2004, pp. 24–31.

[27] K. K. Schuler, "Verbnet: A broad-coverage, comprehensive verb lexicon," Ph.D. dissertation, University of Pennsylvania, 2006. [Online]. Available: http://verbs.colorado.edu/~kipper/Papers/dissertation.pdf

[28] B. Levin, *English Verb Classes and Alternations A Preliminary Investigation*. Chicago and London: University of Chicago Press, 1993.

[29] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995. [Online]. Available: http://doi.acm.org/10.1145/219717.219748

[30] A. Björkelund, L. Hafdell, and P. Nugues, "Multilingual semantic role labeling," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, ser. CoNLL '09. Stroudsburg, PA, USA: ACL, 2009, pp. 43–48. [Online]. Available: http://dl.acm.org/citation.cfm?id=1596409.1596416

[31] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[32] T. Mikolov, M. Karafit, L. Burget, J. Cernock, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*, T. Kobayashi, K. Hirose, and S. Nakamura, Eds. ISCA, 2010, pp. 1045–1048.

[33] A. Taylor, M. Marcus, and B. Santorini, "The Penn Treebank: An overview," 2003.

[34] T. Mikolov, "Statistical language models based on neural networks," Ph.D. dissertation, Ph. D. thesis, Brno University of Technology, 2012.

[35] O. Chapelle, B. Scholkopf, and A. Zien, Eds., *Semi-Supervised Learning*. MIT Press, 2006.

Quynh Ngoc Thi Do is a PhD candidate at the Department of Computer Science of KU Leuven, Belgium. She graduated from the Erasmus Mundus Masters program in Language and Communication Technologies in 2012 with a Master degree in Computer Science from the Free University of Bozen-Bolzano, Italy and a Master degree in Cognitive Science from the University of Lorraine, France.

Steven Bethard received a joint Ph.D. in Computer Science and Cognitive Science in 2007 from the University of Colorado, Boulder, CO, USA. He is currently an assistant professor in Computer and Information Sciences at the University of Alabama at Birmingham, AL, USA, where he is director of the Computational Representation and Analysis of Language (CoRAL) laboratory. He previously worked as a postdoctoral researcher at Stanford University's Natural Language Processing group, Johns Hopkins University's Human Language Technology Center of Excellence, KU Leuven's Language Intelligence and Information Retrieval group in Belgium, and the University of Colorado's Center for Language and Education Research.

Marie-Francine Moens is a professor at the Department of Computer Science of KU Leuven, Belgium, where she is a member of the Human Computer Interaction section. She holds a Ph.D. in Computer Science (1999) from this university. She leads the research group Language Intelligence and Information Retrieval (LIIR - http://www.cs.kuleuven.be/groups/liir/). Her main interests are in the domains of human language understanding and of automated content retrieval from texts with a strong emphasis on content models obtained through machine learning techniques.